PATENT

THE UNITED STATES PATENT AND TRADEMARK OFFICE

Applicant	:	Simon C. Steely, Jr., et al.
Serial No.	:	10/760,659
Filing Date	:	January 20, 2004
For	:	SYSTEM AND METHOD FOR NON-MIGRATORY REQUESTS IN A CACHE COHERENCY
Group Art Unit	:	2188
Confirmation No.	:	9866
Examiner	:	Mardochee Chery
Attorney Docket No.	:	200313629-1

Mail Stop Appeal Briefs - Patents
Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

APPEAL BRIEF

Sir:

Pursuant to the Notice of Appeal filed in this case on July 20, 2007, Appellant's representative presents this Appeal Brief.

I. TABLE OF CONTENTS

II. REAL PARTY IN INTEREST

The real party in interest is Hewlett-Packard Development Company, L.P., as indicated by the Assignment recorded January 20, 2004, Reel/Frame: 014919/0272.

III. RELATED APPEALS AND INTERFERENCES

There are no related appeals or interferences.

IV. STATUS OF CLAIMS

Claims 1-32 which are attached in Appendix A, are currently pending in this application. Claims 1, 8-10, 12, 18-19 and 24-32 stand rejected under 35 U.S.C. §103(a) as being unpatentable over U.S. Patent Application Pub. No. 2002/0129211 to Arimilli et al. ("Arimilli") in view of U.S. Patent Pub. No. 6,931,496 to Chen et al. ("Chen"). Claims 11, and 20-23 stand rejected under 35 U.S.C. §103(a) as being unpatentable over Arimilli in view of Chen and further in view of U.S. Patent Pub. No. 6,484,240 to Cypher et al. ("Cypher"). Claims 2-7 and 13-17 are objected to as being dependent upon a rejected base claim, but would be allowable if rewritten in independent form including all the limitations of the base claim and any intervening claims. The rejection of claims 1, 8-12, and 18-32 is appealed.

V. STATUS OF AMENDMENTS

A Final Office Action ("Final Action") was issued for the present application on March 21, 2007. No amendments were made to the claims after the Final Action.

VI. SUMMARY OF THE CLAIMED SUBJECT MATTER

A. Claim 1

One aspect of the present invention, as recited in claim 1, is directed to a system (10 of FIG. 1) comprising a first node (12 of FIG. 1) including data having an associated state, the associated state of the data at the first node being a modified state (Pars. [0020]-[0023]). The system further (10 of FIG. 1) comprises a second node (14 of FIG. 1) operative to provide a non-migratory source broadcast request for the data (Par. [0044]). The first node (12 of FIG. 1) is operative in response to the non-migratory source broadcast request to provide the data to the second node (14 of FIG. 1; Par. [0044]). The first node (12 of FIG. 1) also is operative to transition the associated state of the data at the first node (12 of FIG. 1) from the modified state to an owner state without updating memory (Par. [0044]). The second node (14 of FIG. 1) is operative to receive the data from the first node (12 of FIG. 1) and assign a shared state to an associated state of the data at the second node (14 of FIG. 1; Pars. [0020]-[0044]).

B. Claim 8

Claim 8 is directed to the system (10 of FIG. 1) of claim 1, wherein the first node (12 of FIG. 1) is operative in response to the non-migratory source broadcast request to provide a shared data response to the second node (14 of FIG. 1; Par. [0047]).

C. Claim 9

Claim 9 is directed to the system (190 of FIG. 6) of claim 1, wherein further migration of the data from the second node (192 of FIG. 6) is precluded when the associated state of the data at the second node (192 of FIG. 6) is the shared state (Par. [0077]).

D. Claim 10

Claim 10 is directed to the system (210 of FIG. 6) of claim 1, further comprising at least one other node (194 of FIG. 6) that provides a non-data response to the second node (192 of FIG. 6) in response to the non-migratory source broadcast request from the second node (192 of FIG. 6; Par. [0007). The non-data response indicates that the at least one other node (194 of FIG. 6) does not have a valid copy of the data requested by the second node (Par. [0077]).

E. Claim 11

Claim 11 is directed to the system (100 of FIG. 3) of claim 1, wherein the first node (102 of FIG. 3) defines a first processor and the second node (104 of

FIG. 3) defines a second processor (104 of FIG. 3; Par. [0059]). The first and second processors (102 and 104 of FIG. 3) each having an associated cache (114 of FIG. 3) that comprises a plurality of cache lines (116 of FIG. 3), each cache line (116 of FIG. 3) having a respective tag address that identifies associated data and each cache line having state information that indicates a state of the associated data for the respective cache line (Par. [0063]). The first and second processors (102 and 104 of FIG. 3) are capable of communicating with each other and with a system memory (110 of FIG. 3) via an interconnect (108 of FIG. 3; Par. [0062]). The system (100 of FIG. 3) further comprises a first cache controller (118 of FIG. 3) associated with the first processor (102 of FIG. 3) and a second cache controller associated with the second processor (104 of FIG. 3; Par. [0063]). The first cache controller (118 of FIG. 3) is operative to manage data requests and responses for the associated cache (114 of FIG. 3) of the first processor (102 of FIG. 3), the first cache controller (118 of FIG. 3) effecting state transitions associated with the data in the associated cache (114 of FIG. 3) of the first processor (102 of FIG. 3) based on the data requests and responses for the associated cache of the first processor (102 of FIG. 3; Par. [0063]). The second cache controller is operative to manage data requests and responses for the associated cache of the second processor (104 of FIG. 3), the second cache controller effecting state transitions associated with the data in the associated cache of the second processor (104 of FIG. 3) based on the data

requests and responses for the associated cache of the second processor (104 of FIG. 3).

F. Claim 12

Another aspect of the invention, as recited in claim 12, is directed to a multi-processor network (100 of FIG. 3) comprising memory (110 of FIG. 3) for storing data (Par. [0058]). The system (100 of FIG. 3) also comprises a first processor node (102 of FIG. 3) having a first processor node cache line (116 of FIG. 3) including the data, the first processor node (102 of FIG. 3) cache line having an associated state, the associated state of the first processor node cache line being a modified state (Par. [0056], [0072]). The system (100 of FIG. 3) further comprises a second processor node (104 of FIG. 3) operative to provide a non-migratory source broadcast read request for the data, the second processor node (104 of FIG. 3) having a second processor node cache line with an associated state (Par. [0072]). The first processor node (102 of FIG. 3) is programmed to respond to the non-migratory source broadcast read request of the second processor node (104 of FIG. 3) by providing a shared data response to the second processor node (104 of FIG. 3) and transitioning the associated state of the first processor node (102 of FIG. 3) cache line from the modified state to an owner state without updating the memory (110 of FIG. 3) with the data, the data being stored in the second processor node cache line, the

associated state of the second processor node (104 of FIG. 3) cache line being assigned a shared state (Par. [0072]).

G. Claim 18

Claim 18 is directed to the network (100 of FIG. 3) of claim 12, wherein further migration of the data from the second processor node (60 of FIG. 2) is precluded when in the shared state (Par. [0056]; [0072]).

H. Claim 19

The network (100 of FIG. 3) of claim 12, further comprising at least one other processor node (106 of FIG. 3) that provides a non-data response (*e.g.*, a MISS response, at Par. [0072]) to the second processor node (104 of FIG. 3) in response to the non-migratory source broadcast read request from the second processor node (104 of FIG. 3), the non-data response (*e.g.*, being a MISS response, at Par. [0072]) indicating that the at least one other processor node (106 of FIG. 3) does not have a valid copy of the data requested by the second processor node (60 of FIG. 2; Par. [0056]; [0072]).

I. Claim 20

The network (100 of FIG. 3) of claim 12, wherein the first and second processor nodes (102 and 104 of FIG. 3) each have an associated cache (114 of FIG. 3) that comprises a plurality of cache lines (116 of FIG. 3), each cache line having a respective tag address that identifies associated data and each cache line having state information that indicates a state of the associated data for the

respective cache line (Par. [0063]). The first and second processor nodes (102 and 104 of FIG. 3) being capable of communicating with each other and with the memory (110 of FIG. 3) via an interconnect (108 of FIG. 3; Par. [0062]). The system (100 of FIG. 3) further comprising a first cache controller (118 of FIG. 3) associated with the first processor node (102 of FIG. 3) and a second cache controller associated with the second processor node (104 of FIG. 3; Par. [0063]). The first cache controller (118 of FIG. 3) is operative to manage data requests and responses for the associated cache (114 of FIG. 3) of the first processor node (102 of FIG. 3; Par. [0063]). The first cache controller (118 of FIG. 3) effecting state transitions associated with the data in the cache of the first processor node (102 of FIG. 3) based on the data requests and responses for the associated cache of the first processor node (102 of FIG. 3; Par. [0063]). The second cache controller is operative to manage data requests and responses for the associated cache of the second processor node (104 of FIG. 3). The second cache controller effecting state transitions associated with the data in the associated cache of the second processor node (104 of FIG. 3) based on the data requests and responses for the associated cache of the second processor node (104 of FIG. 3).

J. Claim 21

Still another aspect of the invention, as recited in claim 21 is directed to a computer system (10 of FIG. 1) comprising a source processor (14 of FIG. 1)

having an associated source processor cache (24 of FIG. 1; Par. [0023]). The source processor (14 of FIG. 1) is operative to issue a selected one of a non-migratory source broadcast (XREADN) request for data (Par. [0040]) and a migratory source broadcast (XREADM) request for data (Par. [0044]). The computer system (10 of FIG. 1) also comprises memory (16 of FIG. 1) storing the data (Par. [0021]). The computer further comprises a target processor (12 of FIG. 1) having an associated target processor cache (22 of FIG. 1) with a target processor cache line (116 in FIG. 3) that stores the data, the target processor cache line having an associated state, the associated state of the target processor cache line being a modified state (Par. [0040]; [0061]). The target processor (12 of FIG. 1) is programmed to respond to the XREADN request by providing a shared data (S-DATA) response to the source processor (14 of FIG. 1) and by transitioning the associated state of the target processor (12 of FIG. 1) cache line from the modified state to an owner state without updating the memory (16 of FIG. 1; Par. [0044]; [0072]). The target processor (12 of FIG. 1) is programmed to respond to the XREADM request by providing an ownership data (D-DATA) response to the source processor and by transitioning the associated state of the target processor (12 of FIG. 1) cache line from the modified state to an invalid state without updating the memory (16 of FIG. 1; Par. [0040]; [0070]).

K. Claim 22

The computer (100 of FIG. 3) system of claim 21, wherein the source processor (104 of FIG. 3) further comprises an associated source processor cache (114 of FIG. 3) having a source processor cache line (116 of FIG. 3) for storing the data (Par. [0061]). The source processor cache line (116 of FIG. 3) has an associated state (Par. [0061]). The source processor (104 of FIG. 3) stores the data in the source processor cache line (116 of FIG. 3) and assigns a shared state to the associated state of the source processor (104 of FIG. 3) cache line (116 of FIG. 3) in response to receiving the S-DATA response from the target processor (102 of FIG. 3; Par. [0071]).

L. Claim 23

The computer system (100 of FIG. 3) of claim 21, wherein the source processor (102 of FIG. 3) further comprises an associated source processor cache (114 of FIG. 3) having a source processor cache line (116 of FIG. 3) for storing the data (Par. [0061]). The source processor cache line has an associated state (Par. [0061]). The source processor (104 of FIG. 3) stores the data in the source processor cache line (116 of FIG. 3) and assigns a dirty state to the associated state of the source processor cache line in response to receiving the D-DATA response from the target processor (102 of FIG. 3; Par. [0069]).

M. Claim 24

A system (10 of FIG. 1, 50 of FIG. 2, 100 of FIG. 3, 150 of FIG. 4, 170 of FIG. 5, 190 of FIG. 6) comprising means for broadcasting from a first node (14 of FIG. 1, 60 of FIG. 2, 104 of FIG. 3, 192 of FIG. 6), a non-migratory read (XREADN) request for data (Pars. [0044], [0056], [0072], [0077]). The system also comprises means for providing the data from a second node (12 of FIG. 1, 56 of FIG. 2, 102 of FIG. 3, 156 of FIG. 4, 196 of FIG. 6) to the first node (14 of FIG. 1, 60 of FIG. 2, 104 of FIG. 3, 192 of FIG. 6) in response to the XREADN request, a modified state being associated with the data at the second node (Pars. [0044], [0056], [0072], [0077]). A shared state is associated with the data at the first node (14 of FIG. 1, 60 of FIG. 2, 104 of FIG. 3, 192 of FIG. 6) in response to the first node (14 of FIG. 1, 60 of FIG. 2, 104 of FIG. 3, 192 of FIG. 6) receiving the data from the second node (12 of FIG. 1, 56 of FIG. 2, 102 of FIG. 3, 156 of FIG. 4, 196 of FIG. 6; Pars. [0044], [0056], [0072], [0077]). The system further comprises means for transitioning (122 of FIG. 3) the modified state associated with the data at the second node (12 of FIG. 1, 56 of FIG. 2, 102 of FIG. 3, 156 of FIG. 4, 196 of FIG. 6) to an owner state without updating memory (16 of FIG. 1, 72 of FIG. 2, 110 of FIG. 3, 198 of FIG. 6) of the system (10 of FIG. 1, 50 of FIG. 2, 100 of FIG. 3, 150 of FIG. 4, 170 of FIG. 5, 190 of FIG. 6; Pars. [0044], [0056], [0072], [0077]).

N. Claim 25

Claim 25 is directed to the system (10 of FIG. 1, 50 of FIG. 2, 100 of FIG. 3, 150 of FIG. 4, 170 of FIG. 5, 150 of FIG. 4) of claim 24, further comprising means for broadcasting from the first node (14 of FIG. 1, 60 of FIG. 2, 104 of FIG. 3, 152 of FIG. 4) a migratory read (XREADM) request for data (Pars. [0040], [0054], [0070], [0075]). The system still further comprises means for providing the data from the second node (12 of FIG. 1, 54 of FIG. 2, 102 of FIG. 3, 156 of FIG. 4) to the first node (14 of FIG. 1, 60 of FIG. 2, 104 of FIG. 3, 152 of FIG. 4) in response to the XREADM request, the modified state being associated with the data at the second node (12 of FIG. 1, 54 of FIG. 2, 102 of FIG. 3, 156 of FIG. 4; Pars. [0040], [0054], [0070], [0075]). The shared state is associated with the data at the first node (14 of FIG. 1, 60 of FIG. 2, 104 of FIG. 3, 152 of FIG. 4) in response to the first node (14 of FIG. 1, 60 of FIG. 2, 104 of FIG. 3, 152 of FIG. 4) receiving the data from the second node (12 of FIG. 1, 54 of FIG. 2, 102 of FIG. 3, 156 of FIG. 4; Pars. [0040], [0054], [0070], [0075]). The system (10 of FIG. 1, 50 of FIG. 2, 100 of FIG. 3, 150 of FIG. 4, 170 of FIG. 5, 150 of FIG. 4) also comprises means for transitioning the modified state associated with the data at the second node to an invalid state without updating memory (16 of FIG. 1, 72 of FIG. 2, 110 of FIG. 3, 158 of FIG. 4) of the system (10 of FIG. 1, 50 of FIG. 2, 100 of FIG. 3, 150 of FIG. 4, 170 of FIG. 5, 150 of FIG. 4; Pars. [0040], [0054], [0070], [0075]).

O. Claim 26

Claim 26 is directed to the system of claim 25, further comprising means for selecting (12 of FIG. 1) one of the XREADM request and XREADN request to broadcast from the first node (14 of FIG. 1, Par. [0034]).

P. Claim 27

Claim 27 is directed to the system of claim 25, further comprising means for predictively selecting (12 of FIG. 1) one of the XREADM request and XREADN request to broadcast from the first node (14 of FIG. 1, Par. [0034]).

Q. Claim 28

Still yet another aspect of the invention is directed to a method comprising broadcasting (300 of FIG. 8) a non-migratory request for data from a first node to other nodes of an associated system (Par. [0079]). The method also comprises providing (310 of FIG. 8) a shared copy of the data from a second node to the first node in response to the non-migratory request (Par. [0079]). The method further comprises transitioning (320 of FIG. 8) a state associated with the data at the second node from a modified state to an owner state in response to the non-migratory request (Par. [0079]). The method still further comprises transitioning (330 of FIG. 8) a state associated with the data at the first node to a shared state in response to receiving the shared copy of the data from the second node (Par. [0079]).

R. Claim 29

Claim 29 is directed to the method of claim 28, further comprising broadcasting a migratory request for the data from the first node (14 of FIG. 1) to other nodes (12 of FIG. 1) of the associated system. The method also comprises providing an ownership data response from the second node (12 of FIG. 1) to the first node (14 of FIG. 1) in response to the migratory request (Par. [0040]). The method also comprises transitioning the state associated with the data at the second node (12 of FIG. 1) from a modified state to an invalid state data in response to the migratory request (Par. [0040]). The method also comprises transitioning the state associated with the data at the first node (14 of FIG. 1) to a dirty state in response to receiving the ownership data response from the second node (Par. [0040]).

S. Claim 30

Claim 30 is directed to the method of claim 29, further comprising selecting one of the migratory request and the non-migratory request to broadcast from the first node (14 of FIG. 1; Par. [0034]).

T. Claim 31

Claim 31 is directed to the method of claim 29, further comprising predictively selecting one of the migratory request and the non-migratory request to broadcast from the first node (14 of FIG. 1; Par. [0034]).

U. Claim 32

Even yet another aspect of the invention, as recited in claim 32, is directed to a computer system (10 of FIG. 1) comprising a cache coherency protocol that is operative to permit migration of data to a cache (24 of FIG. 1) associated with a source processor (14 of FIG. 1) from a cache (22 of FIG. 1) associated with a target processor (12 of FIG. 1) when a migratory request is issued from the source processor (12 of FIG. 1; Par. [0042]). The protocol is operative to prevent migration of the data to the cache (24 of FIG. 1) associated with the source processor (14 of FIG. 1) from the cache (22 of FIG. 1) associated with the target processor (12 of FIG. 1) when a non-migratory request is issued from the source processor (14 of FIG. 1; Par. [0046]).

VII. GROUNDS OF REJECTION TO BE REVIEWED ON APPEAL

A. Whether claims 1, 8-10, 12, 18-19 and 24-32 are made obvious by Arimilli in view of Chen.

B. Whether claims 11 and 20-23 are made obvious by Arimilli in view of Chen and further in view of Cypher.

VIII. ARGUMENT

A. 35 U.S.C. §103(a) Rejection of Claims 1, 8-10, 12, 18-19 and 24-32 as being Made Obvious by Arimilli in view of Chen

To reject claims in an application under section 103 of Title 35, an examiner must show an unrebutted *prima facie* case of obviousness. See *In re Deuel*, 51 F.3d 1552, 1557, 34 U.S.P.Q.2d 1210, 1214 (Fed. Cir. 1995). In the absence of a proper prima facie case of obviousness, an applicant who complies with the other statutory requirements is entitled to a patent. See *In re Oetiker*, 977 F.2d 1443, 1445, 24 U.S.P.Q.2d 1443, 1444 (Fed. Cir. 1992).

The following objective inquiry is to control the analysis under 35 U.S.C. 103:

> "Under §103, the scope and content of the prior art are to be determined; differences between the prior art and the claims at issue are to be ascertained; and the level of ordinary skill in the pertinent art resolved. Against this background the obviousness or nonobviousness of the subject matter is determined. Such secondary considerations as commercial success, longfelt but unsolved needs, failure of others, etc., might be utilized to give light to the circumstances surrounding the origin of the subject matter sought to be patented." *KSR v. Teleflex*, 550 U.S. ___, 127 S. Ct. 1727 (2007), citing *Graham v. John Deere Co. of Kansas City*, 383 U. S. 1 at 17–18 (1966).

1. The Obviousness Rejection of Claims 1 and 28

In the rejection of claim 1, it is contended in the Final Action that Arimilli teaches a second node, as recited in claim 1 (See Final Action, Page 11 citing Page 7 of Arimilli). Appellant's representative respectfully disagrees. In rejecting

claim 1, the Final Action purports to separate interrelated features and alleges support for a first set of features are found in Arimilli and that support for a second set of features are found in a second reference. However, this approach fails to give proper consideration to the claim as a whole. The portions of claim 1 which the Final Action considers to be disclosed in Arimilli must be considered in view of the other claim language, such as including the state of the data at the first node and the particular type of request provided by the second node. The separate reliance on Arimilli appears to give insufficient weight and even ignores the interrelationships between the state of data at the nodes and the requests and responses recited in the system of claim 1.

Contrary to the contentions in the Final Action regarding claim 1, Arimilli fails to teach or suggest that the second node is operative to receive data from the first node and that the second node is operative to assign a shared state to an associated state of the data at the second node, as recited in claim 1. The Detailed Action in the Final Action relies entirely on the teachings at page 7 of Arimilli to support the rejection of claim 1. However, Arimilli discloses arbitrating between conflicting requests to modify a cache line that is held in a shared state and for protecting ownership of the cache line granted during such arbitration (See Arimilli, Page 7, Par. [0060]). Nothing on page 7 of Arimilli (or anywhere else in Arimilli) teaches or suggests a second node receives data from a first node and assigning a shared state to an associated state of the data at the

second node, such as recited in claim 1. In Arimilli, if the coherency state associated with a target cache line has a state other than shared or invalid (including a modified state), a master 26 performs a store into cache array 24 without issuing a transaction on the system bus (See Arimilli, Par. [0036]). The master 26 in Arimilli prevents access to the target cache line by other agents 10 by means of appropriate snoop responses until the store into the cache array is completed (See Arimilli, Par. [0036]).

In Arimilli, if the target cache line is invalid, the master 26 issues a read with intent to modify (RWITM) transaction on the system bus 12 to obtain a copy of the cache line from another agent 10 for modification (See Arimilli Par. [0036]). If the master 26 receives a combined response awarding ownership of the target cache line for purposes of modifying the cache line (OWNER CR), the master 26 performs a store (of the target cache line) because all of the agents 10 caching the same (target) cache line have invalidated or will invalidate their respective copies of the cache line (See Arimilli, Par. [0057] and Table III). If the agent 10 receives a combined response indicating that the master 26 has awarded ownership of the target cache line for purposes of modifying the cache line, but must perform clean up operations to maintain coherency (OWNER_CU CR), the master 26 issues high priority kill transactions on the system bus 12 before performing the store (See Arimilli, Par. [0058], Table III and FIG. 3B). The clean up operations, which are utilized to maintain coherency, are described in Arimilli

in invalidating the cache line for all agents (including any previous owner), such that in contrast to claim 1 any prior owner who might have held the data in the modified state before the transaction now holds the data in the invalid (I) state (See Arimilli, Par. [0058], [0060]).

Additionally, according to Arimilli, the agent that issues the read with intent to modify (RWITM) transaction seeks and is provided ownership, if the CDP grants ownership, such that the agent that issues the transaction (Dclaim or RWITM) does not store data in the shared state, but instead is awarded ownership (Arimilli, at Par. [0057], [0061].

The Response to Arguments section of the Final Action contends that paragraph [0029] of Arimilli also teaches certain features of claim 1; namely that a “second node operative to receive data from a first node and that the second node is operative to assign a shared state to an associated state of the data at the shared node,” such as recited in claim 1 (See Final Action, Page 2). Again, this contention fails to appreciate the particular interrelated set of circumstances recited in claim 1 that provide for and enable this result. Appellant's representative respectfully submits that paragraph [0029] of Arimilli relates to an identification of a situation wherein a conflict can arise (e.g., between times t0 and t2, See Arimilli, Par. [0029] and FIG. 2) when a modifying transaction has been issued for a target cache line that is marked as shared in another agent’s cache. Thus, Appellant's representative respectfully submits that paragraph

[0029] is unrelated to the subject matter recited in claim 1. Furthermore, Appellant's representative asserts that in none of the situations taught or suggested by Arimilli describes a second node receives data from a first node having data in a modified state and assigning a shared state to the data at the second node, consistent with the system recited in claim 1. This is because in the approach disclosed in Arimilli, a target cache line is either: (a) already in the shared state (thus the data is not provided by another node) or (b) received from another agent 10, wherein, when the target cache line is received, the agent 10 is awarded (exclusive) ownership (as indicated by the OWNER CR or OWNER_CU CR) by the master 26, such that the target cache line is not assigned a shared state.

The Final Action admits Arimilli is deficient as it "does not explicitly teach the first node being operative in response to the non-migratory source broadcast request to provide the data to the second node and to transition the associated state of the data at the first node from the modified state to an owner state without updating system memory (Final Action, at page 12, first paragraph)." Chen fails to make up for the deficiencies of Arimilli.

In contrast to the contentions in the Final Action, none of the sections of Chen cited in the Final Action (e.g., Cols. 3, 5 and 6 and the Abstract of Chen), nor Chen more generally, teach or suggest that a first node is operative to provide the data to the second node and transition the associated state of the

data at the first node from the modified state to an owner state, in response to a non-migratory source broadcast request provided by the second node as recited in claim 1. The systems and methods of Chen relate to and describe a directory based approach and not a source broadcast based system (See Chen, Col 2, Lines 14 to 25 and Col. 4, Lines. 51-55). Thus, the transactions for cached data in Chen, all go through a distributed shared memory (DSM) controller, such that no non-migratory source broadcast request would exist in the system of Chen. In Chen, the DSM controller maintains and manages states of the data (in an associated directory) for data in a level 3 (L3) cache based on the commands issued by the respective processors (See Chen, Col. 2, lines 20-25). Since no non-migratory source broadcast request exists in Chen, there is no basis to conclude that the teachings of Chen would be operative to provide data to the second node in response to such a non-migratory broadcast request. Therefore, Arimilli taken in view of Chen fails to teach or suggest that a first node is operative, in response to a non-migratory source broadcast request, to provide data to a second node, as recited in claim 1.

Furthermore, Arimilli taken in view of Chen fails to teach or suggest that, in response to the non-migratory source broadcast request, a first node transitions the state of data at such node from a modified state to an owner state without updating system memory, as recited in claim 1. In rejecting claim 1, the Final Action contends that Chen discloses this element of claim 1 (See Final Action,

Page 12). Appellant's representative respectfully disagrees. Chen discloses seven possible states for each of the L3 cache lines: CLEAN, FRESH, DIRTY-ONLY, DIRTY-SHARED, VOID, IDLE (See Chen Col. 4, Lines 60-65). Of the L3 cache states disclosed in Chen, only the DIRTY-ONLY and DIRTY-SHARED states indicate that data has been modified (See Chen Col. 5, Lines 6 to 14). Chen fails to teach or suggest that any command, (even a BRIL command, which is a command issued to read an exclusive copy of the specific data, See Chen Col. 5, Lines 32-33), which enables a second node to receive data from a L3 cache having the data in either of the DIRTY-ONLY and DIRTY-SHARED states and then sets the state of all nodes to a shared state (Chen, Col. 7, lines 48-53). As discussed above, the approach in Chen employs directories in a DSM controller to manage all requests and the states are not associated with data at more than one node, but instead the states apply to the data in the L3 cache of the local node. Accordingly, Chen fails to teach or suggest transitioning from a modified state to an owner state in response to the non-migratory source broadcast request, as recited in claim 1, since in Chen, no command is taught or suggested that enables a second node to receive data from a node with data in either DIRTY-ONLY or DIRTY-SHARED states and then set the state to a shared state.

The Response to Arguments section of the Final Action further cites paragraphs [0037]-[0039] of the present application's Specification and relates

the present application's Specification to various sections of Chen (See Final Action, Pages 3-7). It is noted that the features upon which the Final Action relies (e.g., Spec. Pars. [0037]-[0039]) are not recited in claim 1. For example, paragraph [0037] of Appellant's specification describes transitioning the cache line of the source processor form the I-state to the D-state, whereas claim 1 recites that the second node assigns a shared state to the data. Additionally, paragraph [0038] relates generally to scenario that includes a migratory read request for data, whereas claim 1 explicitly recites that the second node provides a non-migratory source broadcast request for the data. Accordingly, the cited sections of Appellant's specification have been erroneously relied on in the Final Action.

Appellant's representative respectfully submits that the Final Action appears to be impermissibly reading limitations into claim 1 that do not exist, and then basing its rejection on the erroneous limitations that are not present in claim 1. Although the claims are interpreted in light of the specification, limitations from the specification are not read into the claims. See *In re Van Geuns*, 988 F.2d 1181, 26 USPQ2d 1057 (Fed. Cir. 1993). Moreover, Appellant's representative respectfully submits the Final Action has failed to cite any section of Chen (or any other reference) that teaches a first node being operative to respond to a non-migratory source broadcast request, as recited in claim 1. As discussed above, the approach taught in Chen relates to a directory based approach that employs

a DSM controller to process transactions for an L3 cache and not based on a source broadcast request for data, as recited in claim 1.

Further still, the Court of Customs and Patent Appeals has held that if a proposed modification or combination of the prior art would change the principle of operation of the prior art invention being modified, then the teachings of the references are not sufficient to render the claims prima facie obvious. *In re Ratti*, 270 F.2d 810, 813, 270 F.2d 810 (C.C.P.A. 1959). Appellant's representative asserts that one of ordinary skill in the art would not combine and modify the teachings of Chen comprising a DSM system with the system of Arimilli comprising a central memory system because the person or ordinary skill would recognize that the transactions required to maintain coherency in a DSM system would adversely affect the principal operation of a central memory system (Arimilli's System Memory 12; See Chen FIG. 1 and Arimilli Fig 1). Moreover, the combined teachings of Arimilli and Chen still fail to teach or suggest all features recited in claim 1, such that claim 1 would not be obvious to one of ordinary skill in the art.

Moreover, the Final Action contends that one of ordinary skill in the art would see to modify Arimilli to include the transitioning of states from a modified state to an owner state without updating system memory because this would have provided a data maintenance method in a distributed shared memory system to efficiently solve the access deadlock problem as taught by Chen (Final

Action, page 12, line 16, through page 13, line 2). However, the approach in Arimilli is directed to a specific approach in which a coherency decision point (CDP) is utilized as a mechanism to arbitrate between conflict requests to modify data (Arimilli, Par. [0011], [0030]). It is respectfully submitted, that the approach taught in Chen is not consistent with and would likely compromise the intent of the system of Arimilli. The contention to combine the teachings of Arimilli and Chen appear to be unsupported by the record and appear to be based improperly on speculation or the Examiner's subjective belief. *In re Lee*, 277 F.3d 1338, 61 U.S.P.Q.2d 1430 (Fed. Cir 2002).

For these reasons, withdrawal of the rejection of claims 1 and 28 is respectfully requested.

2. <u>The Obviousness Rejection of Claim 8</u>

Claim 8 depends from claim 1 and is patentable for at least the same reasons as claim 1, and for the following reasons. In the rejection of claim 8, it is contended in the Final Action that Chen discloses the elements of claim 8 (See Final Action, Page 13). Appellant's representative respectfully disagrees. Chen fails to teach or suggest that a first node is operative, in response to a non-migratory source broadcast request, to provide a shared data response to a second node, as recited in claim 8. In fact, Chen fails to teach or suggest a source broadcast request - it employs a DSM controller with a directory for

managing data in an L3 cache - such that the system disclosed in Chen is not operative to respond to non-migratory source broadcast requests, in contrast to the first node recited in claim 8. Accordingly, Arimilli taken in view of Chen fails to teach or suggest the system recited in claim 8 since Chen fails to teach or suggest a source broadcast system. Therefore, withdrawal of the rejection of claim 8 is respectfully requested.

3. The Obviousness Rejection of Claim 9

Claim 9 depends from claim 1 and is patentable over Arimilli taken in view of Chen for at least the same reasons as claim 1, and for the specific elements recited therein. Accordingly, withdrawal of the rejection of claim 9 is respectfully requested.

4. The Obviousness Rejection of Claim 12

Arimilli taken in view of Chen fails to make claim 12 obvious. In the Final Rejection, the rationale for the rejection of claim 1 is relied on as the sole basis for the rejection of claim 12 (See Final Action, Page 13). Because of differences between claim 12 and claim 1, the Final Action has failed to present a prima facie case of unpatentability for claim 12. For instance, claim 12 explicitly recites first and second processor nodes having cache lines with associated cache states, and that the second processor node is operative to provide a non-migratory read

request for data, which are not explicitly recited in claim 1. Since Appellant has complied with the other statutory requirements, the absence of a prima facie case of obviousness requires that the rejection of claim 12 be withdrawn. See *In re Oetiker*, 977 F.2d 1443, 1445, 24 U.S.P.Q.2d 1443, 1444 (Fed. Cir. 1992).

Moreover, Arimilli taken in view of Chen fails to teach or suggest data (requested by a non-migratory source broadcast read request) is stored in a second processor node cache line, wherein an associated state of the second processor node cache line is assigned a shared state, as recited in claim 12. In contrast to claim 12, Arimilli teaches that a target cache line is already in a shared state (such that data is not provided by another node) or the target cache line is received from another agent 10, wherein the ownership of the target cache line is awarded by a master 26 to a requesting agent 10, such that the target cache line would not be assigned a shared state (See Arimilli, FIGs. 3A and 3B). Therefore, Arimilli taken in view of Chen fails to teach or suggest the second processor node operating in the manner recited in claim 12.

Moreover, Arimilli taken in view of Chen fails to teach or suggest a first processor node that is programmed to respond to a non-migratory source broadcast read request of the second processor node by providing a shared data response to the second processor node, as recited in claim 12. The Final Action contends that Chen discloses this element of claim 12 (See Final Action, Page 12, in rejecting claim 1), but in contrast to claim 1, Chen teaches a directory

based cache coherency system for an L3 cache that is managed by the a DSM controller (See Chen, Col. 2, Lines 14 to 24 and Col. 4, Lines 51-55). The set of requests and responses taught in Chen relate to requests issued to and responses from the DSM controller for data in the L3 cache. Chen also fails to teach the use of any non-migratory source broadcast read request issued by a source processor node, as recited in claim 12. Accordingly, Arimilli taken in view of Chen cannot teach or suggest that any response (form a first node) to a non-migratory source broadcast read request can be provided to a second node, since Chen fails to teach or suggest a source broadcast based system.

Additionally, Arimilli taken in view of Chen fails to teach or suggest transitioning an associated state of the first processor from a modified state to an owner state without updating the memory, as recited in claim 12. In Chen, which the Final Action contends discloses this element of claim 12 (See Final Action, Page 12), only the DIRTY-ONLY and DIRTY-SHARED states indicate that data in an L3 cache has been modified (See Chen Col. 5, lines 6 to 14). Chen fails to teach or suggest any command that enables a second processor node to receive data from a processor node with the data in either of the DIRTY-ONLY and DIRTY-SHARED states and then set the state to a shared state. Instead, Chen teaches that the cache states and transitions thereof relate to the cache states of the L3 cache (See Chen at Col. 4, line 60, through Col. 5, line 21), which are not cache states for data stored in the cache line of a processor node, as recited in

claim 12. Since Chen fails to teach or suggest a command that enables a second processor node to receive data from a processor node with the data in either of the DIRTY-ONLY and DIRTY-SHARED states and then set the state to a shared state, Arimilli taken in view of Chen also fails to teach or suggest transitioning the associated state of the first processor from a modified state to an owner state without updating the memory, as recited in claim 12.

Furthermore, combining and modifying the teachings of Arimilli and Chen in the manner suggested in the Final action would change a basic principle of operation of Arimilli's central memory system for the reasons stated above with respect to claim 1. Additionally, there is not proper motivation to combine Arimilli and Chen for the reasons discussed above with respect to claim 1. Motivation to modify Arimilli based on teachings in Chen to provide the system of claim 12 is further lacking since Chen relates to management of an L3 cache by DSM controller whereas claim 12 recites first and second processor nodes having cache lines and associated cache states.

For the reasons stated above, Appellant's representative respectfully requests that the rejection of claim 12 be withdrawn.

5. <u>The Obviousness Rejection of Claim 18</u>

Claim 18 depends from claim 12 and is patentable over Arimilli taken in view of Chen for at least the same reasons as claim 12, and for the specific

elements recited therein. Accordingly, withdrawal of the rejection of claim 18 is respectfully requested.

6. The Obviousness Rejection of Claims 10 and 19

Claims 10 and 19 depend from claims 1 and 12, respectively, and are patentable over Arimilli taken in view of Chen for at least the same reasons as claims 1 and 12, and for the following reasons. In rejecting claim 19, the Final Action relies solely on the basis for the rejection of claim 10 (See Final Action, Page 13). In rejecting claim 10, it is contended in the Final Action that Col. 7, Lines 55 to 61 of Chen discloses the elements recited in claim 10 (See Final Action, Page 13). Appellant's representative respectfully disagrees. The cited section of Chen is unrelated to a non-data response that is provided to a second (processor) node, in contrast to the system recited in claims 10 and 19.

Moreover, Chen fails to teach or suggest a relationship between three nodes (or processor nodes, as recited in claim 19): first, second and at least one other node - and also fails to teach or suggest a system with source broadcast requests, consistent with what is recited in claims 10 and 19. Specifically, by virtue of claims 10 and 19's dependence from claims 1 and 12, claims 10 and 19 recite a second node that broadcasts a non-migratory source request for data, a first node that provides the data to the second node, and at least one other node (e.g., a third node) that provides a non-data response to the second node in

response to the non-migratory source broadcast request from the second node, the non-data response indicating that the at least one other node does not have a valid copy of the data requested by the second node. Additionally, in claim 19, the nodes are specifically recited as being processor nodes having cache lines.

In contrast to what appears is being suggested in the Final Action, the commands LRIL or LIL are not responses to non-migratory source broadcast requests, but instead, LRIL and LIL are commands issued to read data (See Chen, Col. 5, Lines 48-50). Significantly, no responses of any kind are disclosed as being issued by any nodes in the cited section of Chen (e.g., Col. 7, Lines 55 to 61 of Chen). Regarding claim 19, Therefore, Appellant's representative respectfully submits that Arimilli taken in view of Chen fails to make claims 10 and 19 obvious since Arimilli taken in view of Chen fails to teach or suggest the interrelationship between the first, second and third node (e.g., the at least one other node) recited in claims 10 and 19. Additionally, in claim 19, the non-data response is provided by a processor node.

For these reasons, withdrawal of the rejection of claims 10 and 19 is respectfully requested.

7. <u>The Obviousness Rejection of Claim 24</u>

Arimilli taken in view of Chen fails to make claim 24 obvious. In the Final Rejection, the rationale for the rejection of claim 1 is relied on as the sole basis

for the rejection of claim 24 (See Final Action, Page 14). Arimilli taken in view of Chen fails to teach or suggest data (requested by a non-migratory source broadcast read request (XREADN)) is provided to a first node, wherein a shared state is associated with the data at the first node in response to the first node receiving the data from a second node, as recited in claim 24. In contrast to the system recited in claim 24, Arimilli teaches that a target cache line is already in a shared state (such that data is not provided by another node) or the target cache line is received from another agent 10, wherein the ownership of the target cache line is awarded by a master 26 to a requesting agent 10, such that the target cache line would not be assigned a shared state (See Arimilli, FIGS. 3A and 3B). Therefore, Arimilli taken in view of Chen fails to teach or suggest the means for providing recited in claim 24.

Moreover, Arimilli taken in view of Chen fails to teach or suggest means for providing data from a second node to the first node in response to a XREADN request, as recited in claim 24. In Chen, which the Final Action contends discloses this element of claim 24 (See Final Action, Page 12, in rejecting claim 1), a directory based cache system is disclosed (See Chen, Col. 2, Lines 14 to 24 and Col. 4, Lines 51-55) in which a DSM controller manages states of cache lines in an L3 cache. In contrast to claim 24, Chen is silent on the employment of any source broadcast based system. Accordingly, Arimilli taken in view of Chen cannot teach or suggest that any response (from a first node) to an XREADN can

be provided to a second node since Chen fails to teach or suggest a source based broadcast system. Therefore, Arimilli taken in view of Chen fails to teach or suggest the means for providing recited in claim 24.

Additionally, Arimilli taken in view of Chen fails to teach or suggest means for transitioning a modified state associated with data at a second node to an owner state without updating the memory, as recited in claim 24. In Chen, which the Final Action contends discloses this element of claim 24 (See Final Action, Page 12), only the DIRTY-ONLY and DIRTY-SHARED states indicate that data in an L3 cache has been modified (See Chen Col. 5, Lines 6 to 14). Chen fails to teach or suggest any non-migratory command that would enable a second node to receive data from a node with the data in either of the DIRTY-ONLY and DIRTY-SHARED states and then set the state to a shared state upon receipt, while also transitioning the data in either of the DIRTY-ONLY and DIRTY-SHARED states to an owner state, as recited in claim 24.

Furthermore, combining and modifying the teachings of Arimilli and Chen in the manner suggested in the Final Action would change a basic principle of operation of Arimilli's central memory system (System Memory 24) for the reasons discussed above with respect to claim 1. Additionally, it would not be reasonable for one of ordinary skill in the art to modify Arimilli in view of the teachings of Chen to provide the system of claim 24 for the reasons discussed above with respect to claim 1.

For these reasons, Appellant's representative respectfully requests that the rejection of claim 24 be withdrawn.

8. The Obviousness Rejection of Claims 25-27

Claims 25-27 depend from claim 24 and are patentable over Arimilli taken in view of Chen for at least the same reasons as claim 24, and for the following reasons. In rejecting claim 25 (from which claims 26-27 depend), the Final Rejection relies solely on the rejection of claim 1 (See Final Action, Page 14). Appellant's representative respectfully submits that because of differences between claims 25-27 and claim 1, the Final Action has failed to establish a prima facie case of obviousness with respect to claims 25-27. For instance, claim 25 explicitly recites "means for providing the data from the second node to the first node in response to a migratory read request for data..," whereas claim 1 explicitly recites a scenario that includes a non-migratory read request. Claim 25 also recites "means for transitioning the modified state associated with the data at the second node to an invalid state…" whereas claim 1 recites a scenario where the modified state transitions to an owner state. Since Appellant has complied with the other statutory requirements, the absence of a prima facie case of obviousness in the Final Action requires that the rejection of claims 25-27 be withdrawn. See *In re Oetiker*, 977 F.2d 1443, 1445, 24 U.S.P.Q.2d 1443, 1444 (Fed. Cir. 1992).

In rejecting claim 1, the Final Action has cited no structure or process that could correspond to the XREADM request for data recited in claim 25. In fact, claim 2 (which was indicated as allowable) recites a migratory source broadcast for data. Claims 26 and 27 depend from claim 25 and recite means for selecting one of the XREADM and the XREADM requests, similar to allowable claims 6 and 7, respectively. Accordingly, Appellant's representative respectfully submits that claims 25-27 are patentable for reasons similar to claim 2, 6 and 7. Therefore, the rejection of claims 25-27 should be withdrawn.

9. The Obviousness Rejection of Claims 29-31

Claims 29-31 depend from claim 28 and are patentable over Arimilli taken in view of Chen for at least the same reasons as claim 28 (discussed relative to claim 1), and for the following reasons. Appellant's representative respectfully submits that the Final Rejection of claims 29-31 fails to establish (or even attempts to establish) a prima facie case of obviousness with respect to claims 29-31. In rejecting claim 29 (from which claims 30-31 depend), the Final Rejection relies solely on the rejection of claim 1 (See Final Action, Page 14). However, claim 29 recites broadcasting from a first node a migratory read request for data and the related acts that occur responsive to the migratory request for the data. In rejecting claim 1, the Final Action has cited no function or process that could correspond to the broadcast migratory read request for data

recited in claim 29. In fact, claim 2 (which was indicated as allowable) recites a migratory source broadcast for data. For these reasons and those discussed with respect to claims 25-27, Appellant's representative respectfully submits that claims 29-31 are patentable. Therefore, the rejection of claims 29-31 should be withdrawn.

11. The Obviousness Rejection of Claim 32

Appellant's representative respectfully submits that the Final Rejection of claim 32 fails to establish (or even attempts to establish) a prima facie case of obviousness with respect to claim 32. In rejecting claim 32, the Final Action relies on the rejection of claim 1 as the sole rationale for rejecting claim 32 (See Final Action, Page 14). Appellant's representative respectfully submits that because of differences between claim 32 and claim 1, the Final Action has failed to establish a prima facie case of obviousness with respect to claim 32. For instance, claim 32 explicitly recites “a cache coherency protocol that permits migration of data to a cache associated with a source processor from a cache associated with a target processor when a migratory request is issued..,” whereas claim 1 explicitly includes no mention of such a scenario. Instead a similar recitation of features is found in dependent claim 2, which has been indicated as being allowable. Since Appellant has complied with the other statutory requirements, the absence of presenting a prima facie case of

obviousness for claim 32 requires that the rejection of claims 32 be withdrawn. See *In re Oetiker*, 977 F.2d 1443, 1445, 24 U.S.P.Q.2d 1443, 1444 (Fed. Cir. 1992).

B. 35 U.S.C. §103(a) Rejection of Claims 11 and 20-23 as being made obvious by Arimilli in view of Chen and further in view of Cypher

1. The Obviousness Rejection of Claims 11 and 20

Claims 11 and 20, depend from claims 1 and 12 respectively, and are patentable for at least the same reasons as claims 1 and 12 and for the specific elements recited therein. In rejecting claims 11 and 20, the Final Action relies on Cypher solely for Cypher's disclosure of caches in multiples processor systems that include cache tags for each cache line (See Cypher, Col. 1, Lines 21-25). However, the addition of Cypher does not make up for the deficiencies of Arimilli taken in view of Chen, as discussed above with respect to claims 1 and 12, from which claims 11 and 20 depend. Accordingly, withdrawal of the rejection of claims 11 and 20 is respectfully requested.

2. The Obviousness Rejection of Claim 21

Claim 21 is not made obvious by Arimilli taken in view of Chen and in further view of Cypher. In rejecting claim 21, the Final Action contends that claim 21 is unpatentable over Arimilli, in view of Chen and further in view of

Cypher by incorporating the rationale used in the rejection of claims 1 and 11. Appellant respectfully disagrees with this contention. Appellant's representative respectfully submits that the Final Rejection of claim 21 fails to establish a prima facie case of obviousness with respect to claim 21. See *In re Oetiker*, supra. Claim 21 recites subject matter not recited explicitly in either claims 1 and 11. In particular, claim 21 recites a source processor having an associated source processor cache, the source processor being operative to issues a selected one of a non-migratory source broadcast (XREADN) request for data and a migratory source broadcast (XREADM) request for the data. Appellant's representative respectfully submits that claim 21 recites elements similar to claim 2, which depends from claim 1 and was indicated as being allowable. Accordingly, Appellant's representative respectfully submits that claim 21 is patentable for reasons similar to claim 2.

For these reasons, withdrawal of the rejection of claim 21 is respectfully requested.

3. The Obviousness Rejection of Claims 22-23

Claims 22-23 depend from claim 21 and are patentable over Arimilli taken in view of Chen and in further view of Cypher for at least the same reasons as claim 21, and for the specific elements recited therein. Accordingly, withdrawal of the rejection of claims 22-23 is respectfully requested.

IX. APPENDICES

The first attached Appendix contains a copy of the claims on appeal.

The second and third Appendices have been included to comply with statutory requirements.

No additional fees should be due for this Brief. In the event any fees are due in connection with the filing of this document, the Commissioner is authorized to charge those fees to Deposit Account No. 08-2025.

Respectfully submitted,

/Gary J Pitzer/

Gary J. Pitzer
Registration No. 39,334
Attorney for Appellant(s)

I hereby certify that this correspondence is being transmitted to the U.S. Patent and Trademark Office via electronic filing on September 12, 2007.

CUSTOMER NO.: 022879

Hewlett-Packard Company
Legal Department MS 79
3404 E. Harmony Road
Ft. Collins, CO 80528

Claims Appendix

1. (Finally Rejected) A system comprising:

a first node including data having an associated state, the associated state of the data at the first node being a modified state; and

a second node operative to provide a non-migratory source broadcast request for the data, the first node being operative in response to the non-migratory source broadcast request to provide the data to the second node and to transition the associated state of the data at the first node from the modified state to an owner state without updating memory, the second node being operative to receive the data from the first node and assign a shared state to an associated state of the data at the second node.

2. (Finally Rejected) The system of claim 1, wherein the second node is further operative to provide a migratory source broadcast request for the data, the first node being operative in response to the migratory source broadcast request to provide the data to the second node and to transition the associated state of the data at the first node from the modified state to an invalid state without updating memory, the second node being operative to receive the data from the first node and assign the associated state of the data at the second node to a dirty state.

3. (Finally Rejected) The system of claim 2, wherein the first node is operative in response to the migratory source broadcast request to provide an ownership data response to the second node.

4. (Finally Rejected) The system of claim 2, wherein the associated state of the data at the second node being the dirty state makes the data at the second node available for migration to other nodes.

5. (Finally Rejected) The system of claim 2, wherein the second node is further operative to write the data to the second node and transition the associated state of the data at the second node from the dirty state to the modified state, the associated state of the data at the second node being the modified state making the data at the second node available for migration to other nodes providing a migratory source broadcast request for the data.

6. (Finally Rejected) The system of claim 2, wherein the second node is programmed with instructions to selectively invoke one of the non-migratory source broadcast request and the migratory source broadcast request.

7. (Finally Rejected) The system of claim 2, wherein the second node is programmed with instructions that provide a predictive selection to invoke one of the non-migratory source broadcast request and the migratory source broadcast request.

8. (Finally Rejected) The system of claim 1, wherein the first node is operative in response to the non-migratory source broadcast request to provide a shared data response to the second node.

9. (Finally Rejected) The system of claim 1, wherein further migration of the data from the second node is precluded when the associated state of the data at the second node is the shared state.

10. (Finally Rejected) The system of claim 1, further comprising at least one other node that provides a non-data response to the second node in response to the non-migratory source broadcast request from the second node, the non-data response indicating that the at least one other node does not have a valid copy of the data requested by the second node.

11. (Finally Rejected) The system of claim 1, wherein the first node defines a first processor and the second node defines a second processor, the first and second processors each having an associated cache that comprises a plurality of cache lines, each cache line having a respective tag address that identifies associated data and each cache line having state information that indicates a state of the associated data for the respective cache line, the first and second processors being capable of communicating with each other and with a system memory via an interconnect, the system further comprising a first cache controller associated with the first processor and a second cache controller associated with the second processor, the first cache controller being operative to manage data requests and responses for the associated cache of the first processor, the first cache controller effecting state transitions associated with the data in the associated cache of the first processor based on the data requests and responses for the associated cache of the first processor, the second cache controller being operative to manage data requests and responses for the associated cache of the second processor, the second cache controller effecting state transitions associated with the data in the associated cache of the second processor based on the data requests and responses for the associated cache of the second processor.

12. (Finally Rejected) A multi-processor network comprising:

memory for storing data;

a first processor node having a first processor node cache line including the data, the first processor node cache line having an associated state, the associated state of the first processor node cache line being a modified state; and

a second processor node operative to provide a non-migratory source broadcast read request for the data, the second processor node having a second processor node cache line with an associated state;

the first processor node being programmed to respond to the non-migratory source broadcast read request of the second processor node by providing a shared data response to the second processor node and transitioning the associated state of the first processor node cache line from the modified state to an owner state without updating the memory with the data, the data being stored in the second processor node cache line, the associated state of the second processor node cache line being assigned a shared state.

13. (Finally Rejected) The network of claim 12, wherein the second processor node is further operative to provide a migratory source broadcast read request for the data, the first processor node being programmed to respond to the migratory source broadcast read request of the second processor node by

providing an ownership data response to the second processor node and transitioning the associated state of the first processor node cache line from the modified state to an invalid state without updating the memory with the data, the data from the ownership data response being stored in the second processor node cache line and the state associated with the second processor node cache line being assigned a dirty state.

14. (Finally Rejected) The network of claim 13, wherein the data stored in the second processor node cache line assigned the dirty state is available for migration to other nodes.

15. (Finally Rejected) The network of claim 13, wherein the second processor node is further operative to write the data stored in the second processor node cache line assigned the dirty state and transition the state associated with the second processor node cache line from the dirty state to the modified state, the data stored in the second processor node cache line assigned the modified state being available for migration to other nodes providing a migratory source broadcast read request for the data.

16. (Finally Rejected) The network of claim 13, wherein the second processor node is programmed with instructions to selectively invoke one of the non-

migratory source broadcast read request and the migratory source broadcast read request to obtain the data.

17. (Finally Rejected) The network of claim 13, wherein the second processor node is programmed with instructions to predictively select one of the non-migratory source broadcast read request and the migratory source broadcast read request to obtain the data.

18. (Finally Rejected) The network of claim 12, wherein further migration of the data from the second processor node is precluded when in the shared state.

19. (Finally Rejected) The network of claim 12, further comprising at least one other processor node that provides a non-data response to the second processor node in response to the non-migratory source broadcast read request from the second processor node, the non-data response indicating that the at least one other processor node does not have a valid copy of the data requested by the second processor node.

20. (Finally Rejected) The network of claim 12, wherein the first and second processor nodes each have an associated cache that comprises a plurality of cache lines, each cache line having a respective tag address that identifies

associated data and each cache line having state information that indicates a state of the associated data for the respective cache line, the first and second processor nodes being capable of communicating with each other and with the memory via an interconnect, the system further comprising a first cache controller associated with the first processor node and a second cache controller associated with the second processor node, the first cache controller being operative to manage data requests and responses for the associated cache of the first processor node, the first cache controller effecting state transitions associated with the data in the cache of the first processor node based on the data requests and responses for the associated cache of the first processor node, the second cache controller being operative to manage data requests and responses for the associated cache of the second processor node, the second cache controller effecting state transitions associated with the data in the associated cache of the second processor node based on the data requests and responses for the associated cache of the second processor node.

21. (Finally Rejected) A computer system comprising:

a source processor having an associated source processor cache, the source processor being operative to issue a selected one of a non-migratory source broadcast (XREADN) request for data and a migratory source broadcast (XREADM) request for data;

memory storing the data; and

a target processor having an associated target processor cache with a target processor cache line that stores the data, the target processor cache line having an associated state, the associated state of the target processor cache line being a modified state, the target processor being programmed to respond to the XREADN request by providing a shared data (S-DATA) response to the source processor and by transitioning the associated state of the target processor cache line from the modified state to an owner state without updating the memory, the target processor being programmed to respond to the XREADM request by providing an ownership data (D-DATA) response to the source processor and by transitioning the associated state of the target processor cache line from the modified state to an invalid state without updating the memory.

22. (Finally Rejected) The computer system of claim 21, wherein the source processor further comprises an associated source processor cache having a source processor cache line for storing the data, the source processor cache line having an associated state, the source processor storing the data in the source processor cache line and assigning a shared state to the associated state of the source processor cache line in response to receiving the S-DATA response from the target processor.

23. (Finally Rejected) The computer system of claim 21, wherein the source processor further comprises an associated source processor cache having a source processor cache line for storing the data, the source processor cache line having an associated state, the source processor storing the data in the source processor cache line and assigning a dirty state to the associated state of the source processor cache line in response to receiving the D-DATA response from the target processor.

24. (Finally Rejected) A system comprising:

means for broadcasting from a first node a non-migratory read (XREADN) request for data;

means for providing the data from a second node to the first node in response to the XREADN request, a modified state being associated with the

data at the second node, a shared state being associated with the data at the first node in response to the first node receiving the data from the second node; and

means for transitioning the modified state associated with the data at the second node to an owner state without updating memory of the system.

25. (Finally Rejected) The system of claim 24, further comprising:

means for broadcasting from the first node a migratory read (XREADM) request for data;

means for providing the data from the second node to the first node in response to the XREADM request, the modified state being associated with the data at the second node, the shared state being associated with the data at the first node in response to the first node receiving the data from the second node; and

means for transitioning the modified state associated with the data at the second node to an invalid state without updating memory of the system.

26. (Finally Rejected) The system of claim 25, further comprising means for selecting one of the XREADM request and XREADN request to broadcast from the first node.

27. (Finally Rejected) The system of claim 25, further comprising means for predictively selecting one of the XREADM request and XREADN request to broadcast from the first node.

28. (Finally Rejected) A method comprising:

broadcasting a non-migratory request for data from a first node to other nodes of an associated system;

providing a shared copy of the data from a second node to the first node in response to the non-migratory request;

transitioning a state associated with the data at the second node from a modified state to an owner state in response to the non-migratory request; and

transitioning a state associated with the data at the first node to a shared state in response to receiving the shared copy of the data from the second node.

29. (Finally Rejected) The method of claim 28, further comprising:

broadcasting a migratory request for the data from the first node to other nodes of the associated system;

providing an ownership data response from the second node to the first node in response to the migratory request;

transitioning the state associated with the data at the second node from a modified state to an invalid state data in response to the migratory request; and

transitioning the state associated with the data at the first node to a dirty state in response to receiving the ownership data response from the second node.

30. (Finally Rejected) The method of claim 29, further comprising selecting one of the migratory request and the non-migratory request to broadcast from the first node.

31. (Finally Rejected) The method of claim 29, further comprising predictively selecting one of the migratory request and the non-migratory request to broadcast from the first node.

32. (Finally Rejected) A computer system comprising a cache coherency protocol that is operative to permit migration of data to a cache associated with a source processor from a cache associated with a target processor when a migratory request is issued from the source processor, the protocol being further operative to prevent migration of the data to the cache associated with the source processor from the cache associated with the target processor when a non-migratory request is issued from the source processor.

Evidence Appendix

None

Related Proceedings Appendix

None